Seismic Inversion for Fracture Model Reconstruction: From 1D Inversion to Machine Learning

Maxim Protasov[1(✉)], Roman Kenzhin[1,2], Danil Dmitrachkov[1], and Evgeniy Pavlovskiy[2]

[1] Institute of Petroleum Geology and Geophysics, Koptyug St. 3, Novosibirsk 630090, Russia
protasovmi@ipgg.sbras.ru
[2] Novosibirsk State University, Pirogova St. 2, Novosibirsk 630090, Russia

Abstract. The presented paper is devoted to the numerical study of the applicability of 1D seismic inversion and 2D machine learning based inversion for fracture model reconstruction. Seismic inversion is used to predict reservoir properties. Standard version is based on a one-dimensional convolutional model, but real geological media are more complex, and therefore it is necessary to determine conditions where seismic inversion gives acceptable results. For this purpose, the work carries out a comparative analysis of one-dimensional and two-dimensional convolutional modeling. Also, machine learning methods have been adopted for 2D fracture model reconstruction. We use UNet architecture and 2D convolutional model to create a training dataset. We perform numerical experiments for a realistic synthetic model from Eastern Siberia and Sigsbee model.

Keywords: seismic inversion · convolution model · fractures · machine learning

1 Introduction

To date, seismic inversion has become a standard procedure, which is used to make a detailed model of the elastic parameters of the geological environment, which enables to predict reservoir properties [1, 2]. Researchers have defined seismic inversion as an algorithm that converts a time or migrated time seismic section into an acoustic impedance section [1, 3]. In this sense, the input data for the inversion are the seismic data time processing results, both before and after stacking [4, 5]. However, in areas with complex geology, seismic depth processing is required to build an accurate representation of the reservoir structure. In this case, the image after depth migration is transformed into the time domain, and standard inversion algorithms are applied.

But seismic inversion is based on the use of a one-dimensional convolutional model assuming horizontal layering of the medium [1, 3]. In practice, it may not satisfy this condition due to the presence of various kinds of lateral inhomogeneities, for example fractures provide them. In this case, it is necessary to switch to two-dimensional or three-dimensional convolutional models and implement the corresponding inversion

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
O. Gervasi et al. (Eds.): ICCSA 2023, LNCS 13957, pp. 99–109, 2023.
https://doi.org/10.1007/978-3-031-36808-0_7

algorithms. However, this does not happen in practice, and for laterally inhomogeneous media, seismic inversion is used, which is implemented within the framework of a one-dimensional convolutional model. Therefore, in the presented work, we determine the conditions where a one-dimensional seismic inversion gives acceptable results, and where it does not, through a numerical study on model examples. To do this, we compare the numerical results of one-dimensional and two-dimensional convolutional modeling, and we also compare the numerical results of seismic inversion. We carry the study out using the synthetic model Sigsbee.

At present, the problem of identifying complex carbonate reservoirs and predicting their porosity and permeability properties according to seismic data does not have an unambiguous solution [6, 7]. The irregular distribution of fractures complicates the structure of such reservoirs, which, in turn, are the main ways of fluid filtration and form the basis of the capacitive space of fractured reservoirs. 1D seismic inversion does not help in this case, therefore, as we mentioned above, there is a necessity to provide 2D and 3D inversion algorithms. On the other hand, machine learning methods shows a great potential in a solution to such kind of problems including 1D seismic inversion [8, 9]. Therefore, in this we adopt the machine learning methods for 2D fracture model reconstruction. For that, we use UNet neural network [10] and 2D convolutional modeling. We perform numerical experiments for a realistic synthetic model from Eastern Siberia.

2 Convolutional Modeling and Seismic Inversion

2.1 1D and 2D Convolutional Models

Acoustic seismic inversion is based on a one-dimensional convolutional model [1, 3]. This formulation suggests that for a horizontally layered model with a normal incidence of a plane longitudinal wave, we can represent a reflected wave field in the time domain as a convolution of the reflection coefficient trace with a seismic impulse:

$$S(t) = w(t) * R(t). \tag{1}$$

Here $S(t)$ – seismic trace, $R(t)$ – reflection coefficients, $w(t)$ – seismic impulse. In the frequency domain, expression (1) is rewritten as follows:

$$S(t) = \int \hat{W}(\omega)\hat{R}(\omega)e^{i\omega t}d\omega, \tag{2}$$

where ω is the frequency, $\hat{W}$ and $\hat{R}$ are the Fourier transforms of the seismic wavelet and reflection coefficients, respectively. Seismic inversion is performed to recover acoustic impedances directly related to reflection coefficients, where the results of time processing are input seismic traces [1, 3]. Also, in practice, an approach is often used when the image after depth migration is transformed into the time domain and standard inversion algorithms based on one-dimensional convolutional modeling are applied.

However, in the case where the medium is not horizontally layered, the results of both time and depth processing in the 2D/3D environment formally satisfy the 2D/3D

convolutional model. Considering the two-dimensional case, let us turn to the result of pre-stack depth migration in true amplitudes. We know that the depth image has an expression that is very similar to a two-dimensional convolution in the frequency domain [11, 12]:

$$S(x_i, z_i) = \iint \hat{W}(\omega(p_x, p_z)) \cdot \hat{R}(p_x, p_z) \cdot e^{i(p_x x_i + p_z z_i)} dp_x dp_z, \tag{3}$$

where p_x, p_z are spatial frequencies, $\hat{W}$ is the Fourier transform of the seismic wavelet, $\hat{R}$ is the Fourier transform of the linearized reflection coefficient $R(x, z)$ in the two-dimensional medium. Similarly to the one-dimensional case, considering the normal incidence of the wave, the time frequency ω depends on the spatial frequencies as follows [11, 12]:

$$\omega(p_x, p_z) = \frac{\sqrt{p_x^2 + p_z^2}}{2} c_0(x_i, z_i), \tag{4}$$

where c_0 is the migration velocity model of the medium. The image, which is got by using the direct calculation of the integral (3), we call the ideal image. Also, it is called as the result of two-dimensional convolutional modeling.

Further, in order to understand where the results of one-dimensional and two-dimensional convolutional modeling are similar and where they differ, we will compare them using specific examples. Since two-dimensional images are defined on a depth scale, and one-dimensional convolution is defined on a time scale, for their comparison, we convert depth images to a time scale using the depth-to-time transform [13].

First, we consider a horizontally layered model, and we calculate seismic traces for it by both one-dimensional and two-dimensional convolutional modeling (see Fig. 1a, 1b). In the first case, we convert the model to the time scale, and we implement one-dimensional convolution according to formula (1). In the second case, we construct images according to formula (3), and then we transform to the time scale by applying the same transform. We calculate the difference between the results as the error of the 1D simulation relative to the 2D simulation (see Fig. 1c). For the horizontally layered medium, the difference between the results of 1D and 2D modeling is mostly about 2–3%. This is partly due to the numerical errors in the approximation of the corresponding integrals. Therefore, we can say that for the model with horizontal interfaces, 1D and 2D modeling give approximately the same results.

Further, we consider a more complex media, i.e. a fragment of the Sigsbee model containing lateral inhomogeneities associated with the presence of non-horizontal layers, faults, and scattering objects. We carry out similar 1D and 2D convolutional modeling computations, and we obtain the corresponding difference in the modeling results (see Fig. 2). Here, we pay attention to the features associated with lateral heterogeneities. One can observe that the results of 1D and 2D convolutional modeling are quite close for part of the model containing interfaces, including those with a non-zero slope angle (the difference is about 2–3%). But in the presence of non-reflecting elements of the geological media such as faults, diffracting or scattering objects, the difference between the modeling results becomes significant, and can reach 40% (see Fig. 2c).

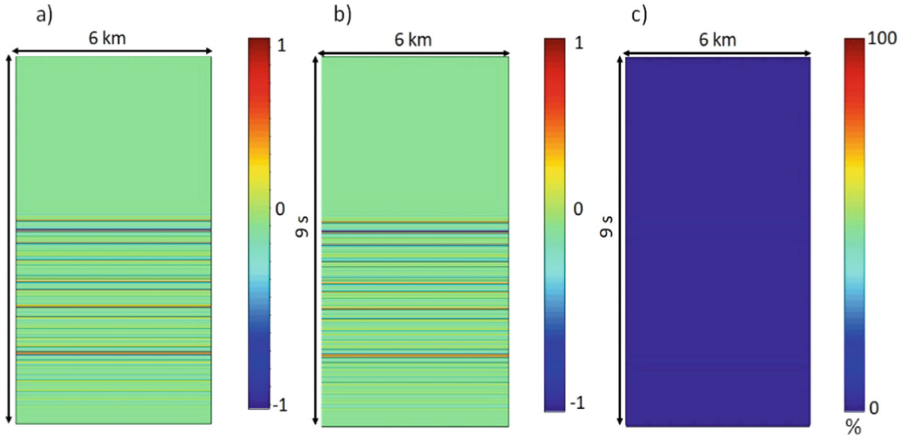

Fig. 1. Results of convolution modeling in the case of a horizontally layered medium: a) 1D modeling; b) 2D modeling; c) their difference.

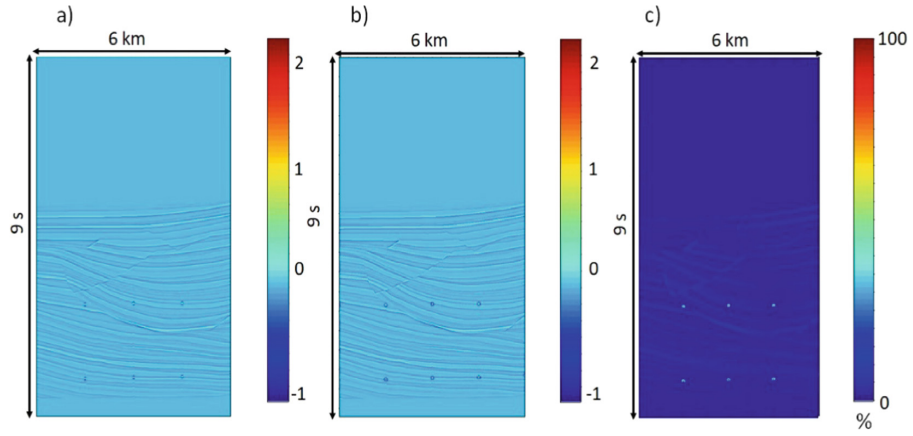

Fig. 2. Results of convolutional modeling in the laterally inhomogeneous Sigsbee model: a) 1D modeling; b) 2D modeling; c) their difference.

2.2 1D Seismic Inversion

One of the most widely used algorithms in practice is model-based inversion. Further, using it, we process the synthetic data for the Sigsbee model got by one-dimensional and two-dimensional convolutional modeling. As shown above, they slightly differ from each other at the interfaces, but significant differences appear in the area where the scattering/diffracting objects are located. The same effect appears when comparing the seismic inversion results (Fig. 3).

For a more detailed comparison, we calculate the error of the inversion results relatively to the true acoustic impedance model (Fig. 4). 1D inversion of 1D modeling data provides the inversion result within the minor error (Fig. 4a). We get such results

because 1D inversion and 1D modeling are inverse operations with respect to each other. However, the results of seismic data processing in a 2D media correspond to 2D convolutional modeling, so it is necessary to track the corresponding error, which arises during 2D seismic data processing.

One can see that the inversion of 2D modeling data provides the error 3–8% in most of the media, and the error can reach 60% in the areas where scatters are located. The incorrect description of essentially two-dimensional effects by the one-dimensional modeling is the reason for such large errors. Experiments show that the application of one-dimensional model-based inversion gives a quite satisfactory result with acceptable accuracy for reflective elements of the medium (Fig. 4b, Fig. 5a). However, the use of seismic inversion in the areas with scattering objects generates significant errors (Fig. 4b, Fig. 5b). This means that 1D seismic inversion must not be applied for reconstruction of scattering objects, particularly for fracture model reconstruction.

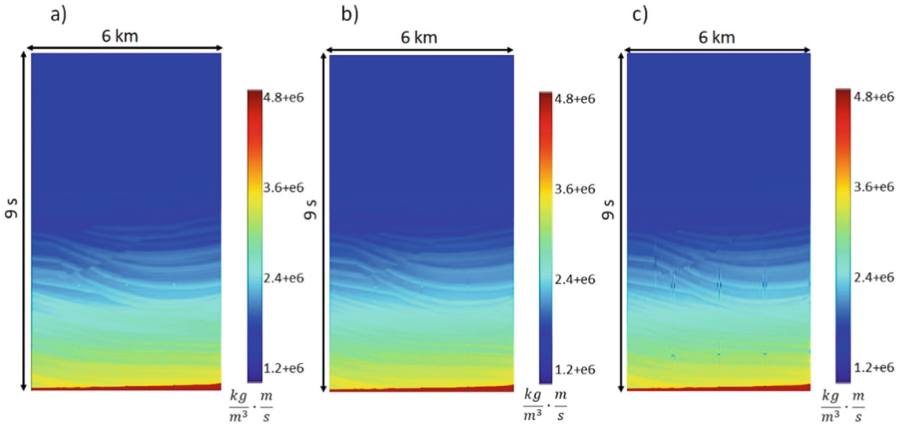

Fig. 3. a) True acoustic impedances. b), c) The result of the 1D inversion according to: b) 1D modeling; c) 2D modeling.

3 Machine Learning for Fracture Model Reconstruction

Based on the described above research results, we suggest investigating the inversion that uses the 2D depth convolutional model. The inversion in this case is not so simple as for 1D time convolutional model. But the 2D convolutional modeling is computationally a cheap procedure. Therefore, we suggest investigating machine leaning methods for 2D fracture model reconstruction.

As one can conclude from the previous section, the 1D seismic inversion provides appropriate results when the model contains reflected interfaces only, i.e. it provides reasonable inversion results of the reflective part of the model. Therefore, 1D seismic inversion can be applied for reconstruction of the reflective part of the model. It means that it is necessary to define the scattering/diffractive part of the model when the model contains the scatters/diffractors.

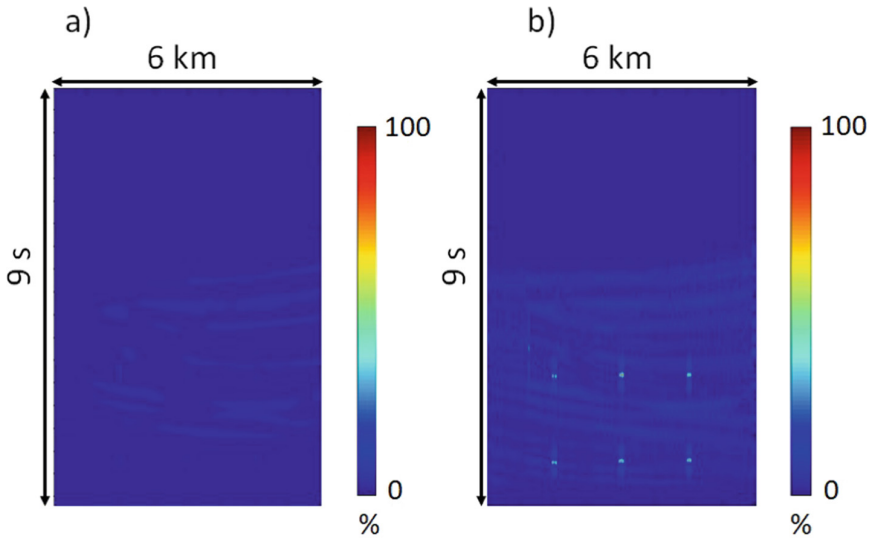

Fig. 4. Relative error of 1D inversion in the Sigsbee model according to: a) 1D simulation; b) 2D modeling.

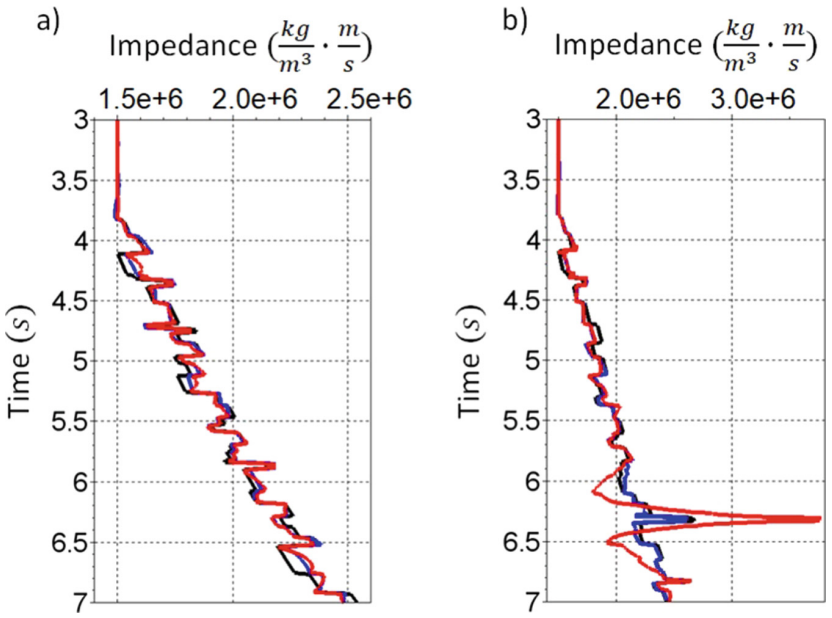

Fig. 5. True acoustic impedances (black), inversion result from 1D modeling data (blue) and 2D modeling data (red): a) trace does not contain scatterers; b) trace contains a scatterer. (Color figure online)

From practical datasets, we know that fractures provide localized heterogeneities of the elastic properties. Therefore, the fractures on the seismic model scale can be considered as scatters/diffractors that are scattering/diffractive part of the model. And this part of the model is the reconstruction goal. On the other hand, we have the result of the seismic depth data processing, which is the depth seismic image. As we said above, this image in the 2D case satisfies 2D convolutional model. Therefore, we create a rather big number of different 2D localized fracture models, then put this fracture model inside reflector model and finally provide the corresponding seismic image. We provide models and images easily using rather small computational resources and minimum a priori information about fractures. Thus, we construct training data set for the machine learning algorithm.

For the described problem, we apply the Unet neural network architecture [10] with changed input and output sizes. We present the architecture at the Table 1.

4 Numerical Experiments

4.1 2D Synthetic Fracture Models and Images

To test the technology, a two-dimensional seismogeological model was created based on a prototype section of one of the East Siberian fields (see Fig. 6a). The elastic model parameters are constructed based on well data. Small-scale heterogeneities typical for this field were included in the model: fractures, located in the upper part of the Riphean carbonate complex (see Fig. 6b). The petrophysical characteristics of heterogeneities were set based on the results of the analysis of the core material. The fractures belong to the class of fluid-saturated ones. We can create a variety of different fracture models with the information from the well data. In this particular case, we construct 1000 models (see one of them on the Fig. 6b). For all of them, we construct corresponding full models (see the corresponding model on the Fig. 7a) including reflective part of the model that is the same for all 1000 fracture models. And finally, we provide 2D convolution modeling in order to get the ideal image (see the corresponding image on the Fig. 6c). Thus, we construct a data set for testing the proposed machine learning approach.

4.2 ML Results of 2D Fracture Model Reconstruction

Then we come to the application of the proposed machine learning approach to the created data. From that dataset, we use 800 realizations for training and 200 realizations for testing of size 2000×400 pixels. We interpret the task in terms of machine learning as a semantic segmentation, i.e. to classify each pixel to belong to a background - 0, or fracture - 1. The splitting is carried out once without shuffling. The neural network is fed by samples without preprocessing. We apply a post-processing min-max image normalization at the output of the neural network. We modify the UNet architecture to work with 2000×400 input-output size (Table 1).

We define the loss function as an equal combination of Dice loss (5), binary cross-entropy (6) and mean squared error losses, which averages on all pixels of an image; $\boldsymbol{y}$

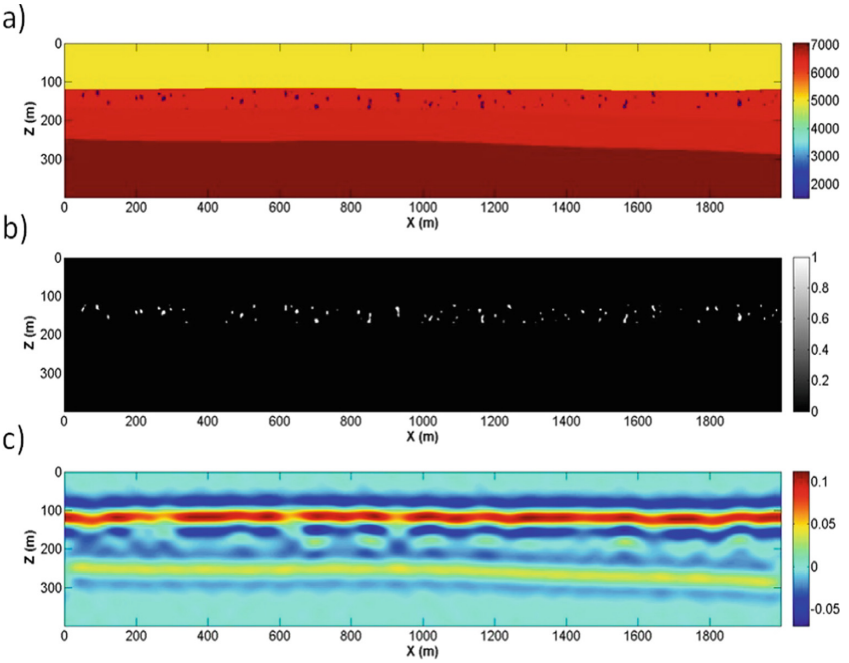

Fig. 6. a) Seismic velocity model (pressure velocity) containing main reflected horizons of Riphean carbonate complex, and one of the fracture model realization. b) The corresponding fracture model realization – indicator function showing the fractures density. c) The corresponding ideal seismic image – the result of 2D convolution modeling.

stands for the true image, and $\hat{\boldsymbol{y}}$ is the predicted by UNet.

$$Dice(y, \hat{y}) = 2\frac{y \cap \hat{y}}{y + \hat{y}}, \; DiceLoss = 1 - Dice, \tag{5}$$

$$BCE\big(y, \hat{y}\big) = -\big[\hat{y} \log y + \big(1 - \hat{y}\big) \log(1 - y)\big] \tag{6}$$

Also, we use the Dice metric to measure the quality of the classifier. Neural network training parameters are the following: optimizer - AdamW, learning rate - 0.1, batch size - 2, maximum number of epochs – 25. We carry out validation on the test dataset at the end of training after the 25th epoch. The total error in the training was 0.62, on the test – 0.74.

After we finish the training, we apply the trained neural network to the images that were used in the training dataset first. One of the images and the corresponding recovered model results one can see on the Fig. 7. Recovered fractures locations have red color, the true fractures locations have blue color (Fig. 7b). There is rather good correspondence between the biggest true fracture locations and the recovered ones. Dice metric in this case is rather high, and it equals 37.0%.

Then we apply the trained neural network to the images from the testing dataset. One of the images and the corresponding recovered model results we can see on the Fig. 8.

Table 1. Network architecture. MP is MaxPooling2D (kernel = 2 × 2, stride = 2, padding = 0, dilation = 1), DC(x,y) is Conv2D(in = x, out = y, kernel = 7 × 7,stride = 1, padding = 3) – BatchNorm – ReLU - Conv2D(in = y, out = y, kernel = 3 × 3, stride = 1, padding = 1) – BatchNorm – ReLU.

Layer structure	Channels	Size
Input	1	2000 × 400
DC(1,64)	64	2000 × 400
MP-DC(64,128)	128	1000 × 200
MP-DC(128,256)	256	500 × 100
MP-DC(256,512)	512	250 × 50
MP-DC(512,1024)	1024	125 × 25
ConvTranspose2d(1024,512,2 × 2, s = 2 × 2)-DC(1024,512)	512	250 × 50
ConvTranspose2d(512,256,2 × 2, s = 2 × 2)-DC(512,256)	256	500 × 100
ConvTranspose2d(256,128,2 × 2, s = 2 × 2)-DC(256,128)	128	1000 × 200
ConvTranspose2d(128,64,2 × 2, s = 2 × 2)-DC(128,64)	64	2000 × 400
Conv2D(64,1,1 × 1,1)	1	2000 × 400
Output – MinMax[0,1]	1	2000 × 400

There is also good correspondence between the biggest true fracture locations and the recovered ones. Dice metric in this case is rather high, and it equals 28.8%.

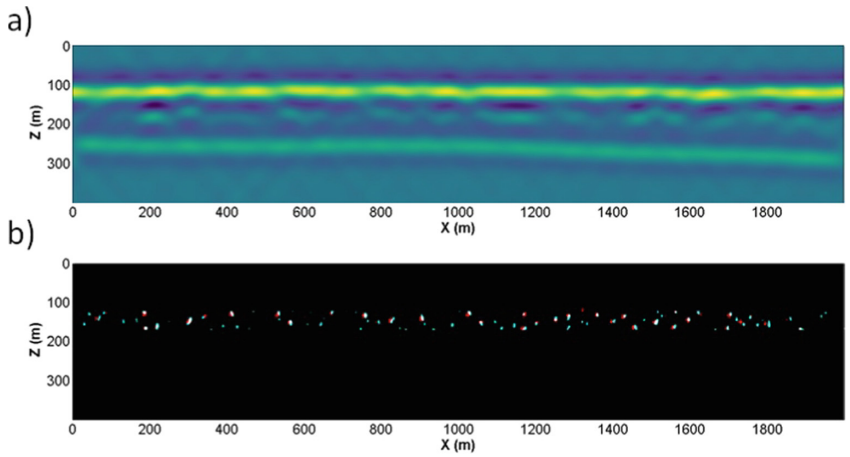

Fig. 7. a) The seismic image from the training dataset. b) The corresponding fracture models: blue – true fractures locations; red – recovered fractures locations. Dice metric equals to 37.0%. (Color figure online)

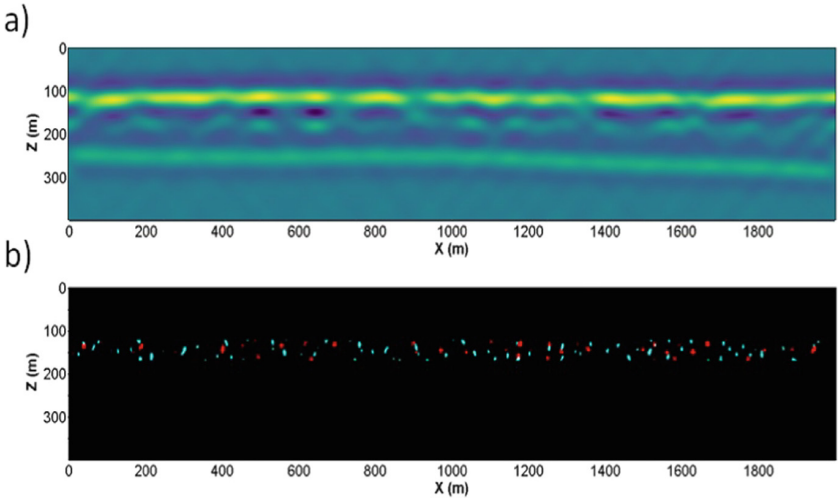

Fig. 8. a) The seismic image from the testing dataset. b) The corresponding fracture models: blue – true fractures locations; red – recovered fractures locations. Dice metric equals to 28.8%. (Color figure online)

5 Conclusions

In the paper we describe 1D and 2D convolution modeling algorithms and the implementation of seismic inversion algorithm for fracture model recovery based on 2D convolution modeling and machine learning.

We provide a numerical study of the applicability of seismic inversion for laterally inhomogeneous media. On 2D synthetic models, we show that the results of 1D convolutional modeling with acceptable accuracy (within 2–3%, including depth-time transformation errors) coincide with the results of 2D convolutional modeling both for models with horizontal reflecting boundaries and for models with boundaries with non-zero dip angle. But when the non-reflecting elements such as scattering objects (caverns, faults, fractures) are present in the media, the difference between the results of one-dimensional and two-dimensional convolutional modeling becomes significant, and can reach 40%. The numerical results of the application of the implemented seismic inversion to the 2D convolutional modeling data show that the error for the main part of the Sigsbee model is acceptable (about 3–8%). In areas of this model where faults and scatters are present, the inversion error becomes significant and reach 60%. This means that 1D seismic inversion must not be applied for reconstruction of scattering objects, particularly for fracture model reconstruction. But the 1D seismic inversion can be used as an effective tool for the recovery of the reflected component of the model.

For the fracture model reconstruction, we provide a machine learning approach. It contains two key components: 2D depth convolution modeling for creating training dataset, and UNet neural network architecture for the fracture model reconstruction. We provide the machine learning results for 2D models and images. The numerical results for realistic synthetic model from the Eastern Siberia show appropriate reconstruction of

the fracture model. These results look promising for the generalization of the approach to 3D real physical models.

Acknowledgements. The work is supported by RSF grant 21-71-20002. The numerical results were obtained using the computational resources of Peter the Great Saint-Petersburg Polytechnic University Supercomputing Center (scc.spbstu.ru).

References

1. Ampilov, Yu.P., Barkov, A.Yu., Yakovlev, I.V., Filippova, K.E., Priezzhev, I.I.: Almost everything is about seismic inversion. Part 1. Seismic Technol. **4**, 3–16 (2009)
2. Yakovlev, I.V., Ampilov, Yu.P., Filippova, K.E.: Almost everything is about seismic inversion. Part 2. Seismic Technol. **1**, 5–15 (2011)
3. Russell, B.H.: Introduction to Seismic Inversion Methods. Course Notes Series, Society of Exploration Geophysicists, pp. 80–101 (1988)
4. Russell, B.H., Hampson, D.P.: Comparison of poststack seismic inversion methods. In: SEG Technical Program Expanded Abstracts, pp. 876–878 (1991)
5. Hampson, D.P., Russell, B.H., Bankhead, B.: Simultaneous inversion of pre-stack seismic data. In: SEG Technical Program Expanded Abstracts, pp. 1633–1637 (2005)
6. Liu, E., et al.: Fracture characterization by integrating seismic-derived attributes including anisotropy and diffraction imaging with borehole fracture data in an offshore carbonate field. In: International Petroleum Technology Conference, IPTC-18533 (2015)
7. De Ribet, B., Yelin, G., Serfaty, Y., Chase, D., Kelvin, R., Koren, Z.: High resolution diffraction imaging for reliable interpretation of fracture systems. First Break **35**(2), 43–47 (2017)
8. Chen, Y., Schuster, G.: Seismic inversion by Newtonian machine learning. Geophysics **85**, W185–W200 (2020)
9. Pintea, S.L., Sharma, S., Vossepoel, F.C., van Gemert, J.C., Loog, M., Verschuur, D.J.: Seismic inversion with deep learning. Comput. Geosci. **26**(2), 351–364 (2021). https://doi.org/10.1007/s10596-021-10118-2
10. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
11. Protasov, M.I., Tcheverda, V.A.: True amplitude imaging. Dokl. Earth Sci. **407**, 441–445 (2006)
12. Protasov, M., Tcheverda, V.: True amplitude imaging by inverse generalized Radon transform based on Gaussian beam decomposition of the acoustic Green's function. Geophys. Prospect. **59**, 197–209 (2011)
13. Robein, E.: Seismic Imaging. EAGE Publications (2010)